

LISTING OF THE CLAIMS

At the time of the Action:

Pending Claims: 1-31 and 33-38

Canceled Claim: 32

After this Response:

Pending Claims: 1-29 and 39-47

Amended Claims: 1, 6, 13, 15-21, and 28-29

Canceled Claims: 30-38

New Claims: 39-47

1. (Currently Amended) A computer readable storage medium encoded with a first data structure and a second data structure, comprising:
a first parameter definition for ~~at least one~~ a first input parameter, the first parameter definition ~~being configured~~ to enable identification of an appropriate first input for the ~~at least one~~ first input parameter, wherein the first parameter definition is a declared property ~~in a declared class~~ of the first data structure; ~~and~~
a second parameter definition for a second input parameter, the second parameter definition to enable identification of an appropriate second input for the second input parameter, wherein the second parameter definition is a declared property of the second data structure; and
an instruction-based mechanism ~~configured~~ to use the first parameter definition to identify the appropriate first input for the ~~at least one~~ first

input parameter, and use the second parameter definition to identify the appropriate second input for the second input parameter from information included in an input source, the input source comprising at least one pipelined live object outputted by a second data structure that is identical to the first data structure,

wherein the instruction-based mechanism is to further configured to enable the first data structure to apply one or more directives associated with the declared class that specify at least one of a machine role or a user role, and process the at least one first input parameter based on the appropriate first input identified from an input source to output an live object, and provide the object as an input for the second input parameter to be processed by the second data structure by passing a reference of the object to the second data structure,

when the first and second data structures become[[s]] instantiated into an objects.

2. (Previously Presented) The computer readable storage medium of claim 1, wherein the input source comprises a string.

3. (Previously Presented) The computer readable storage medium of claim 2, wherein the string comprises a part of a script.

4. (Previously Presented) The computer readable storage medium of claim 2, wherein the string comprises a part of a command string entered on a command line.

5. (Previously Presented) The computer readable storage medium of claim 1, wherein the parameter definition comprises a data type and a name for the expected input parameter.

6. (Currently amended) The computer readable storage medium of claim 1, wherein the input source includes information ~~comprises~~ a value.

7. (Previously Presented) The computer readable storage medium of claim 6, wherein the parameter definition comprises a data type and a name for the expected input parameter, and wherein the mechanism further coerces the value having a first data type into a converted value having a second data type specified in the definition.

8. (Previously Presented) The computer readable storage medium of claim 1, wherein the input source comprises a set of objects.

9. (Previously Presented) The computer readable storage medium of claim 8, wherein the set of objects comprise .NET objects.

10. (Previously Presented) The computer readable storage medium of claim 1, wherein the input source comprises a precisely parseable stream.

11. (Previously Presented) The computer readable storage medium of claim 10, wherein the precisely parseable stream comprises an XML-based document.

12. (Previously Presented) The computer readable storage medium of claim 1, wherein the mechanism further identifies and populates each input parameter for each record within the input source.

13. (Currently Amended) The computer readable storage medium of claim 1, further comprising a mapping mechanism that is operative to associate a mapped name with the first input parameter, ~~wherein identifying the information is based on the mapped name.~~

14. (Previously Presented) The computer readable storage medium of claim 1, wherein the mechanism comprises a method inherited from a class provided within a runtime environment.

15. (Currently Amended) The computer readable storage medium of claim 1, wherein the at least one of first and second data structures is of a public class.

16. (Currently Amended) The computer readable storage medium of claim 1, wherein at least one of the first and second input parameters is a ~~are~~ public parameter[[s]].

17. (Currently Amended) The computer readable storage medium of claim 1, ~~further comprising a plurality of parameter definitions, each parameter definition corresponding to one of a plurality of input parameters,~~ wherein at least one of the parameter definitions ~~is configured to be indirectly associated with~~ one of the first and second data structures.

18. (Currently Amended) The computer readable storage medium of claim 17, wherein the indirect association between the at least one of the parameter definitions and the one of the first and second data structures comprises a reference to an XML-based document that enables identification of the corresponding input parameter.

19. (Currently Amended) A computer-executable method for processing an input source populating parameters declared within a data structure, the method comprising:

~~obtaining an expected name for a parameter, the expected name being assigned via a declared property in a declared class of a data structure;~~
~~identifying a label within an input source correlating to the expected name, the input source comprising at least one live object;~~
~~retrieving a value associated with the label;~~
~~assigning the value to the parameter;~~
~~applying one or more declared directives associated with the declared class of a data structure to the value, the one or more declared directives being associated with the parameter, wherein the one or more declared directives are configured to cause an administrative tool framework to process the parameter~~

retrieving a first parameter definition for a first input parameter, the first parameter definition to enable identification of an appropriate first input for the first input parameter, wherein the first parameter definition is a declared property of a first data structure;

retrieving a second parameter definition for a second input parameter, the second parameter definition to enable identification of an appropriate second input for the second input parameter, wherein the second parameter definition is a declared property of a second data structure;

identifying the appropriate first input from an input source based on the first parameter definition;

processing the first input parameter on the first data structure using the appropriate first input to output an object;

providing the object as an input for the second input parameter by passing a reference of the object to the second data structure; and
processing the object on the second data structure.

20. (Currently Amended) The method of claim 19, wherein the processing the object includes processing the object when the object is identified as the appropriate second input based on the second parameter definition~~the expected name and the label are identical.~~

21. (Currently Amended) The method of claim 19, further comprising providing mapping information that defines an alias name for the first input parameter~~expected name~~ and identifying the first input parameter~~label~~ based on the alias name.

22. (Original) The method of claim 21, wherein the input source comprises a command string entered on a command line and the alias name is provided within the command string.

23. (Original) The method of claim 21, wherein the alias name is provided within a data store.

24. (Original) The method of claim 19, wherein the input source comprises an XML document.

25. (Original) The method of claim 19, wherein the input source comprises a database table.

26. (Original) The method of claim 19, wherein the input source comprises a command string entered on a command line.

27. (Original) The method of claim 19, wherein the input source comprises a script.

28. (Currently Amended) A system ~~to the~~ handle([s]) input parameters, the system comprising:

a means for processing; and

a memory means, the memory means being allocated for a plurality of computer-executable instructions which are loaded into the memory means for execution by the means for processing, the computer-executable instructions performing a method comprising:

~~a means for obtaining an expected name for a parameter, the expected name being assigned via a declared property in a declared class of a data structure;~~

~~a means for identifying a label within an input source correlating to the expected name, the input source comprising at least one live object;~~

~~a means for retrieving a value associated with the label;~~

~~a means for assigning the value to the parameter;~~

~~a means for applying one or more declared directives associated with the declared class of a data structure to the value, the one or more declared directives being associated with the parameter, wherein the one or more declared directives are configured to cause an administrative tool framework to process the parameter~~

a means for retrieving a first parameter definition for a first input parameter, the first parameter definition to enable identification of an appropriate

first input for the first input parameter, wherein the first parameter definition is a declared property of a first data structure;
a means for retrieving a second parameter definition for a second input parameter, the second parameter definition to enable identification of an appropriate second input for the second input parameter, wherein the second parameter definition is a declared property of a second data structure; and
a means for identifying the appropriate first input from an input source based on the first parameter definition,
a means for processing the first input parameter on the first data structure using the appropriate first input to output an object as an input for the second input parameter by passing a reference of the object to the second data structure; and
a means for processing the object on the second data structure when the object is identified as the appropriate second input based on the second parameter definition.

29. (Currently Amended) A computer readable storage medium encoded with a first data structure and a second data structure, that provides a template for creating an application, the data structure comprising:

a name identifying an application that is included in a declared parent class provided by an object-based environment;
at least one member configured to receive one or more sets of input, wherein each set of input comprises at least one live object; and
a method associated with the one or more sets of input,

~~wherein the declared parent class is configured to provide processing that executes the method for each set of input received for the at least one member when the name of the application is invoked, and wherein the application comprises a command in a pipeline of commands, and the set of input comprises results from a previous command in the pipeline of commands~~

a first parameter definition for a first input parameter, the first parameter definition to enable identification of an appropriate first input for the first input parameter, wherein the first parameter definition is a declared property of the first data structure;

a second parameter definition for a second input parameter, the second parameter definition to enable identification of an appropriate second input for the second input parameter, wherein the second parameter definition is a declared property of the second data structure; and

an instruction-based mechanism to use the first parameter definition to identify the appropriate first input for the first input parameter, and use the second parameter definition to identify the appropriate second input for the second input parameter,

wherein the instruction-based mechanism is to further enable the first data structure to process the first input parameter based on the appropriate first input identified from an input source to output an object as an input for the second input parameter, and enable the second data structure to process the object when the instruction-based mechanism identifies the object as the appropriate second input based on the second parameter definition.

30. -38. (Canceled).

39. (New) The computer readable storage medium of claim 29, wherein the input source comprises a string.

40. (New) The computer readable storage medium of claim 40, wherein the string comprises a part of a script.

41. (New) The computer readable storage medium of claim 40, wherein the string comprises a part of a command string entered on a command line.

42. (New) The computer readable storage medium of claim 29, wherein the parameter definition comprises a data type and a name for the expected input parameter.

43. (New) The computer readable storage medium of claim 29, wherein the input source comprises a set of objects.

44. (New) The computer readable storage medium of claim 43, wherein the set of objects comprise .NET objects.

45. (New) The computer readable storage medium of claim 29, wherein the input source comprises a precisely parseable stream.

46. (New) The computer readable storage medium of claim 45, wherein the precisely parseable stream comprises an XML-based document.

47. (New) The computer readable storage medium of claim 29, wherein the mechanism further identifies and populates each input parameter for each record within the input source.